Re:*Port*

Vol. 2, Issue 2 **The Independent Newsletter for Portfolio Users** Nov./Dec., 1992

Universal Interface adds 512K, two ports

BSE has created the first Universal Interface for the Portfolio.

The unit combines a parallel interface and a serial interface in one regulation-sized unit. The unit attaches to the expansion bus and is 2.75'' by 3.75''.

Optionally, you can add 512K to the unit, also!

BSE, well-known in the Port world as makers of the FlashDrive hard disks, has finally created what XoteriX promised , but couldn't deliver, long ago.

And there's more. The BSE Universal Interface includes a ROM chip that works as drive B:, which includes XTERM2, UPDATE, File Manager, and even PBASIC. It also includes the drivers for the FlashDrive hard disks.

Re:*Port* has tested an early version of the unit. This version was sparse on documentation and had a problem with its ROM chip (PBASIC and one of the hard disk utilities didn't work), but BSE is working on solving both problems.

If you know how to make an EEPROM, you can even replace the ROM chip's programs with your own.

Much like a parallel interface, the UI has a 24-pin female connector at the far end. The serial interface portion is a Macintosh-like connector. BSE provides an adapter cable that ends in a 9-pin male connector, just like the standard serial interface.

The interface without additional RAM is only $99.95. The version with 512K of RAM is $249.95, and you can order them both through Re:*Port* newsletter. Please note that although some interfaces have shipped, there may be some shipping delays because of the problems with the internal ROM chip.

The most exciting development, besides having both a serial and a parallel interface combined, is the 512K option.

To install the extra RAM, you simply attach the UI to the expansion bus and reset your Portfolio (by removing the batteries and pressing the silver switch in the compartment). Be sure to back up your C: drive first, because it will be wiped out!

When the Port reboots, you will have 640K of memory available (the Port's 128K plus 512K). You can even make your C: drive up to 592K!

Of course, not everyone needs 512K. Let's dispell a few myths about extra Port memory, and then

Continued on Page 3

What's Inside

• In response to your requests, we have a full back-issue listing. **Page 2**

• BJ Gleason is back with a column about getting the most out of memory with PBASIC. **Page 4**

• Programs, programs, programs! PC Card Drive utilities, an editor hook, a game, and more! **Pages 6-8**

• Win a 64K card! Find out how to enter. **Page 11**

• Beginner's column, advanced column, mailbox, contest winner, and more are all ... **Inside**

The contents of this newsletter and diskette are ©1992 Re:*Port*, except where noted
Portfolio is a trademark of the Atari Corporation

From the Editor

Enhance your Portfolio with Re:*Port* back issues

This is Re:*Port*'s eighth issue, which means that anyone who is still a first-time subscriber doesn't have every issue! We get requests almost every day for contents of issues that people have missed so they can decide which ones to buy.

Back issues are available only to subscribers and are $10 each. Of course, they include the Re:*Port* disk in your preferred format. Please include an extra $2 if you are ordering Volume 1, Issue 2 on 5.25" disks, because it includes an extra disk.

All issues are available, but please be patient if you order one -- if an issue should be out of stock, it will take a few days to get more copies made.

Send your check or money order, along with the numbers of the back issues that you would like to order, to:

Re:*Port* Back Issues
806-34 Benton Drive
Iowa City, IA 52246-5207

Here is a complete list of contents of the newsletters. If there are disk files associated with articles, we have enclosed them in parentheses.

Volume 1, Issue 1

Article: Welcome to Re:*Port*

Gleason on PBasic (PBASIC)

From the Editor: 60View instructions (60VIEW)

Editorial: Non-disclosure policy slows Port development

DOS Tips

Backup and restore C: drive with batch files (BACKUP.BAT, RESTORE.BAT)

Editor: Macro puts text in date, documents (MACRO)

Worksheet: Checkbook balancing (CHECK.WKS)

Address Book: Using the Port as a study partner

Industry News

Volume 1, Issue 2 (includes two 5.25" disks, $2 extra)

Spell checker (SPLCHK)

Stupid Port Tricks: Worksheet games (DICE.WKS)

Quick Tip: Battery watching

Industry News: HyperList, music program, RAM card replacement cases

Gleason on PBasic (PBASIC 4.9)

PBasic vs. PowerBASIC

Online Services (Complete text listings of CompuServe and GEnie library areas. Since then, we have published only bimonthly additions)

Backup/restore (BR)

Book Review: Abacus Port guide

Bug RePort: Macro update (MACRO)

Tip on hiding files

Mailbox

Volume 1, Issue 3

Program Reviews: Finance, Terminator 2 File Manager, PowerBASIC, Business Contacts and Information Manager

Editor: Getting .HOO files to work

Industry news: Port memory, Util compiler, shareware disks

Checkbook program with link to Quicken (CHECK)

Settings program (SETTINGS)

Piano program (PIANO)

Gleason on PBasic

Online Services (LISTINGS, BANNER, CGASHOW, CODE, DIAL, ESHOW, LAUNCH, MAC, MUTE, PGCUNIT, SCRMODE, SECURE, TRIG)

Ordering information

Mailbox (UPDATE.COM, ANSI.SYS)

Volume 1, Issue 4

Updated Launch program (LAUNCH)

Industry News: 512K cards, APB book, graphics service

Gleason on PBasic

Worksheet tip

Review: Instant Speller

Battery monitor program (BAT)

Jeopardy helper program (JEO)

Beginner's column: Dropping File Manager

Online Services (LISTINGS)

Advanced column: Graphics

Looking ahead: Port in space

DOS to Port: Adventure games

Services: Fast shipping, custom programming

Volume 1, Issue 5

Worksheet: Plot graphics (SEEWORK)

Port uses: Police work

Industry News: Hard drives, carrying cases, European software

Review: HyperList

Gleason on PBasic (PREAD)

Graphics programs (PDRAW, PGCVIEW, ANIMATIONS)

Solitaire game (SOL)

Beginner's column: Getting files from the desktop to your Port

Port problems: How to install a PC Card Drive

Quick tip: Faxing

Advanced programming: Graphics routines in Turbo Pascal

User-to-User: Get things done

Mailbox

Continued on Page 5

Industry News

1/2/4 meg FlashRAM cards not available

In very sad news for all Portfolio users, the Optrol company announced that it would be at least one year before they received enough FlashRAM chips to go into production on 1-meg, 2-meg, and 4-meg RAM cards.

Re:*Port* received many orders for the cards. Most people who were good enough to make an order will receive their money back.

However, Optrol was able to use their current stock of chips to make a limited run of 1-megabyte cards only. Re:*Port* expects to receive 10 of these before the end of the month, and all of them are spoken for. If you ordered your card before Sept. 30, you should be receiving one of this batch.

We're very sorry that this venture did not work out because of something as frustrating as a supply problem. However, the cutting-edge of technology always involves some risk, and we're glad that so many Re:*Port* readers and others were so involved in the Port that they felt they needed more storage.

All is not lost, however. Optrol is considering creating 512K cards, which they can produce in quantity within a few months. If you are interested, please contact Optrol at 1-919-779-3377 and let them know that you think it is a good idea. Any expanded storage is better than none, we think. And the price will still be competitive with the price of a 128K card from Atari.

FLOPPY DRIVE AVAILABLE

In an odd turn of events, a device that was built for the HP95LX palmtop computer will work with the Portfolio. It is something that users have been asking for, too: a 3.5" floppy disk drive!

Reader Mel Zwillenberg reports that the Sparcom95 floppy disk drive will work with the Portfolio if you add the following statement to the Port's CONFIG.SYS:

```
DEVICE=DRIVE95.SYS POQET H9600
```

The drive will read 720K and 1.44Meg 3.5" floppies. It connects to a serial interface.

However, the drive will not format MS-DOS disks. The disks must be formatted on another computer. That is rarely a problem, because disks can easily be purchased pre-formatted at most stores.

The drive is available from EduCalc. Call 1-800-677-7001 for details on prices and availability. It includes rechargeable batteries and a charger, but it does not include the serial interface.

RE:*PORT* IN ENGLAND

If you wish to continue your subscription in England, but you don't want to go through the hassle of getting U.S. funds, you now have a choice.

The Portfolio Connection in England will sell you a one-year subscription for 30 pounds, including P&P.

The Portfolio Connection is also an excellent source for shareware Portfolio disks and a free newsletter with Port tips.

For more information, contact:
The Portfolio Connection
Bucklands Cottage
Wallingford Rd.
Cholsey, OX10 9HB
Tel: 0491 36880

Universal Interface

Continued from Page 1

explain the benefits.

First, extra memory will not expand the amount of room avaiable in the internal applications. You are still limited to about 50K worth of data, because of the way the internal applications were written.

Second, you must leave the 512K unit installed. Removing it will destroy your C: drive.

The extra memory, though, will be a boon to those who would like to use the Portfolio to run DOS programs (especially compilers and larger text adventure games). We tested Turbo Pascal 6.0, which died on the unexpanded Port, and it compiled programs just fine with the 512K exapnsion.

If you just use the Port's internal applications, buy the UI without RAM. If you use DOS, go for the UI plus 512K! Either way, you get both interfaces in one.

PBASIC

Make the most of the Port's limited memory

Gleason on PBASIC

ON DISK

PBASIC can be found on Re:*Port* Disk #8 in the \PBASIC subdirectory.

©1992 BJ Gleason

The Portfolio has a limited amount of memory. Although some users have upgraded to 512k, the rest of us are still sitting with about 128k. Well, not really. With an 8k C:, and the files and buffers, the memory drops down to 100k or less.

I won't lie to you. PBASIC requires about 70k or memory just for itself, thus leaving you, the PBASIC programmer, anywhere from 5 to 30k, depending on your memory configuration. Not bad, but what if you want to run programs that are a bit larger than the amount of free memory that you have?

In this column, we will look at various techniques to shrink down the size of PBASIC programs.

LINE NUMBERS, REMARKS and SPACES

In short - get rid of them! Line numbers are only needed as targets of GOTOs and GOSUBs. Get rid of the rest. Use one-digit numbers on the remaining lines.

Remarks are comments for the programmer. Getting rid of them can trim quite a bit of space off the size of the program.

Get rid of excess spaces in the lines. Most PBASIC statements can be written right next to each other, including the space between the line number and the statement. Use can also use the program SPB.EXE which will get rid of the spaces and remarks.

STRINGS

If you are using string variables in your program, you can save memory by using the SSIZE variable. By default, each string is 256 bytes in size, regardless of how much of the string you use. If your program is only using short strings, you can change the default size of the strings. This statement:

```
SSIZE = 40
```

at the beginning of your program, will change the default size of the strings to 40 characters each. This can have quite an effect on memory savings.

COMBINE LINES

You can save a few bytes by combining statement lines with the ':' statement separator, instead of a newline, since the newline is a two byte sequence, and the ':' is only one byte.

SHORT VARIABLE NAMES

While not a something I would teach my students, using single character variable names can trim some space from the program size. Start out using A-Z and then two letter names, and so on.

RUN and CHAIN

If a program is still too big, it might be possible to split the program in several small programs, and run via the RUN statement, or the CHAIN statement. RUN will release all the memory, load and execute the specified program. CHAIN will retain all the variables that have already been defined, so the various segments of code can exchange information.

There is also the system variable called ISRUN, which will be set if the currently running program was started via the RUN statement. This will allow you to add a line like:

```
IF ISRUN THEN RUN "MAIN"
```

to return you to the main program.

SUMMARY

Using these techniques, and others, you can fit 10 pounds of potatoes into a 5 pound bag.

About the Author

Mr. BJ Gleason is an instructor at The American University in the Computer Science and Information Systems Department. He has been programming for more than a decade and is the author of PBASIC 4.9.

You can write to him at:
BJ Gleason
The American University
CSIS (Thin Air Labs)
4400 Massachusetts Avenue, N.W.
Washington, DC 20016
CompuServe : 75300,2517
EMAIL : bjgleas@auvm.american.edu

Review

Terminal+ is good for needed VT-100 features

Terminal+ from Autobyte is a VT-100 terminal emulator for the Portfolio user who needs access to a mainframe computer.

Aside from VT-100 emulation, there is really no reason to choose Terminal+ over any other Portfolio communications package, such as XTERM2, ACOM, or PORTCOM.

However, if you do need VT-100, it is your only choice right now.

The program comes on a blue ROM card that sticks out from the Port's card drive about an inch.

Using the CardCopy routines described in this issue (see page 6), though, you can transfer the program to a RAM card for easier portability.

Besides, the program expects to store its settings on either the C: drive or a RAM card in the A: drive (which is always the better choice), so you'll need to have a RAM card with you anyway.

Once you've started the program, you can use Terminal's excellent phonebook to dial to common numbers, or you can use direct modem commands to dial out.

The Atari key brings up a list of commands, all of which are very straightforward.

You have the option to send and receive files, but only in ASCII mode. There is no Xmodem protocol.

You can also view files before you send them.

Of course, VT-100 is the key to this program.

You can use it to connect to any computer that supports VT-100 emulation (and nearly all business machines do).

VT-100 expects you to have an 80 by 24 screen, at least, so Terminal+ puts your Portfolio into Static mode, so you can scroll on an 80 by 24 window.

Unfortunately, this is a bit awkward, as you must first press Fn-0 to enter scrolling mode, and then there is no hot-key windowing function such as that found in PORTCOM to allow you to go to any area of the screen quickly.

Terminal+ also does not have a ''scroll-back'' buffer to allow you to see what you missed from many screens of data. However, VT-100 is usually sent screen-by-screen, with plenty of time to read the data before moving on.

Otherwise, the VT-100 mode is quite compatible with basic telecommunications.

In summary, Terminal+ is the only choice for VT-100 communications, but it could use some additional features before it becomes the best choice for Portfolio communications.

To get Terminal+, contact Re:*Port* for pricing and availability.

Back issues

Continued from Page 2

Volume 1, Issue 6

Communications (PORTCOM)

From the Editor: Moving to Iowa

Industry News: Port demo, movie star, RAM cards

Gleason on PBasic

Pacman-like game (PORTMAN)

Windows graphics on the Port (WINPGC)

Editor/Diary: To-do program (TODO)

Voice synthesis (HELP, BIRTHDAY)

Best sellers

Beginner's column: Tips!

Advanced column: Communications routines

Online services (LISTINGS)

Mailbox

Volume 2, Issue 1

News on 1/2/4 meg cards

Industry News: Carrying case, more sound, British newsletter

PowerBASIC column (MENU.BAS, PBRUN)

Snake game (SNAKE)

Bug RePort (TODO)

Editor: Delete utility (DELETE.HOO)

Review: DOS Utilities Card (feature complete list of card contents)

Beginner's column: More on editor hooks

Advanced programming: How to write hooks

Online services (LISTINGS)

Mailbox

Volume 2, Issue 2

You're reading it!

Programs

Backup/restore/unprotect memory cards

If you have a PC Card Drive, you'll love the programs in the CARDCOPY subdirectory of Re:*Port* Disk #8.

CREAD and CWRITE are utilities that allow you to take an exact picture of the RAM or ROM card in your drive and store it for safekeeping. You can then put the picture back on the card at any time, in case you lose something important.

These programs require a PC Card Drive to work. They will not run on the Portfolio.

Taking a picture of your RAM or ROM card will also allow you to break the copy protection scheme employed by many card makers. Usually, in order to make sure that the programs on the card cannot be copied, companies will make a change to the first sector of a memory card. The program will check for this change, and if it is not there, the program halts. CREAD and CWRITE will make sure that the change is on the card!

To use CREAD and CWRITE, you must first run CINSTALL.

When you run CINSTALL, it will ask you for the drive letter of your PC Card Drive. You need only type the letter of the drive (i.e. E: would just be E). It will write a file to your disk that tells CREAD and CWRITE where to find the PC Card Drive.

CREAD, obviously, is the program you'd use to take the picture of the card.

You must have a RAM or ROM card in the drive. Then, type CREAD followed by the filename that you wish to store the card's picture in. This file will be saved to your desktop's hard drive. For example: CREAD 128K.IMG would store a picture file called 128K.IMG on your hard drive.

ON DISK

CREAD, CWRITE, and CINSTALL are on Re:*Port* Disk #8 in the \CARDCOPY subdirectory.

This file will be the exact size of your memory card. You can make it much smaller by compressing it with a program such as PKZIP or ARC. It would also be a good idea to put the date in the filename, so you can remember when the picture was taken.

Use CWRITE to write the image back to the card. You must first put a card in the PC Card Drive, and you must also format that card first (use the CFX command on the PC).

Then, type CWRITE followed by the name of the picture file. For example, CWRITE 128K.IMG.

CWRITE will check to be sure the card is inserted, that it is formatted, and that it is the right size to receive the image file. It will then ask you to make sure that you want to overwrite the current data on the card.

Note that CWRITE will wipe out anything on the card. It will restore the card exactly to its previous state.

SUGGESTED USES

The CardCopy utilities are great for very quick backup and restore purposes. Use them only if you plan to archive a card's contents, but not if you only want to save one or two files. You should use CREAD to make picture files of all of your ROM cards to protect yourself against loss.

Use CardCopy if you have a single RAM card that you would like to use for two or more different purposes. For instance, if you have two clients you travel to see on different days, make an image file of each client's data and store it on your desktop computer. Then, use CWRITE to re-create the memory card whenever you go to see each client.

Although Re:*Port* certainly does not encourage copying of copyright material, it is your right to make backup copies of anything you own. We believe it is also your right to move any program you purchase to any media you choose. You can use the CardCopy programs to do this.

For example, the Terminal+ program reviewed in this issue is on a copy-protected 128K ROM card that has an extra piece that sticks out from the side of the Portfolio. The program itself is only about 25K. If you wish, you could CREAD the Terminal+ RAM card, then CWRITE the card's picture onto your own 128K card.

Then, you would simply fill your 128K card with other programs that you'd like to have with you. The Terminal+ copy protection routines will be on the 128K card and will not affect other programs that you may later store on the drive. Please note, though, that should you later format the 128K card, you will have to go through the procedure again.

Programs

Change the sound of the Port's alarm

ALARM is a utility that will allow you to change the sound of the Portfolio's internal alarm.

Now, instead of playing the same boring beep when your Diary appointment comes, you can have the Port play a song!

To use ALARM, copy it onto a RAM card along with the two .ALR files. An .ALR file is an alarm song.

For ease of use, add the line ALARM <filename> (For example, ALARM SAINTS.ALR) to your AUTOEXEC.BAT file. Then reset your computer. Now, every time your Portfolio is reset, the Alarm change will be made.

To hear the new sound, set a Diary alarm for a time near to the current time and wait. Or, you can use the provided program TESTALAR to play the sound.

ON DISK ALARM, SIMON, and 60HOOK are all in their respective directories on Re:*Port* Disk #8

Right now, only two songs are provided: "When the Saints Go Marching In," and "Twinkle, Twinkle Little Star." We'll add more songs in future issues!

To change the song, type ALARM <filename> with the new song name. You do not have to reset the Port for the change to take effect.

If you wish to write your own song, we will soon come out with a song editor. Until then, the format of the song files are as follows:

One byte indicating a note (as described in the Portfolio Technical Reference Guide) is followed by one byte indicating a time in milliseconds. Then, these two bytes are repeated for different notes in the song. At the end of the file, place two bytes with the value 255. The file can have a length of 255 bytes.

Please contact Re:*Port* with song ideas (one idea for the future is to play Happy Birthday on a birthday Diary date!)

Simon says: Good luck beating this hard game!

Simon is a challenging classic game of memory that's simple to play but incredibly hard to beat.

To play, type SIMON. Press a key when you're ready. You will hear a series of file tones, with a number displayed for each one.

The game automatically turns on the NUMLOCK function, so you can use the numeric keypad numbers (the red keys) on the keyboard to play.

You must memorize the sequence of numbers. When they are done displaying, you must type them in exactly. If you get them right, one more is added next time, and the game gets faster. If you don't get them right, you lose!

SIMON can be started at any level up to 25 by typing SIMON x, where x is the level number.

Good luck! Call us if you get past level 10!

60HOOK.HOO gives you a bigger view of your writing

60HOOK is a program similar to the free 60View program that you received for subscribing to Re:*Port*. However, it hooks directly into the Editor.

To install 60HOOK, copy it and the two .DAT files in its directory onto the main directory of your A: drive. They must be in this directory because of the way the Portfolio runs .HOO files.

To run the program, go into the built-in Editor. It works best if you have set your right margin to 60, which is a good setting anyway because it will give you an inch margin on most 10-point printers.

Press Fn-6 to get a menu of .HOO files. Press 6 for 60HOOK, and it will begin displaying your file.

It will pause after every screen; press any key to continue. You may press the ESC key to exit.

When the file has been fully displayed, 60HOOK will beep twice to let you know that the next key you press will take you back to the Editor.

This is a great way to preview how your formatted page will look when you print it out!

Programs

Resize your C: drive without losing any data

Have you ever wished that there was an easier way to change the size of your C: drive than FDISK? A way that could change the size without destroying your data?

Now there is. If you have a copy of XCOPY.COM (available on the DOS Utilities Card or as part of older versions of DOS), you can use RESIZE, a program that controls XCOPY and various Port features to change the C: drive size.

To use RESIZE, you must first copy RESIZE.EXE and XCOPY.COM onto the root directory of your A: drive.

Then, type RESIZE x, where x is any multiple of 8 used to express the new size of the C: drive in kilobytes. For example, RESIZE 16 would create a 16K C: drive.

ON DISK RESIZE and DIET are in their respective directories on Re:*Port* Disk #8

The program checks to make sure that there is enough room on drive A: to automatically back up and restore your C: drive. If there is, it proceeds to copy all files from C: to A:, FDISK your C: drive, reboot the computer, copy the files back to drive C:, erase the files from drive A:, and reboot again.

Part of the magic (and the reason these programs need to be on drive A:'s root directory) is that it creates an AUTOEXEC.BAT file that the Port will read when it reboots after FDISKing the C: drive. This AUTOEXEC file tells the Port restore the C: drive files, erase all the old files from drive A:, erase the AUTOEXEC itself, and reboot the computer again to restore it to its regular state.

Although RESIZE can easily move you from a smaller C: size to a larger one, it cannot guarantee no loss of data if you move from a larger drive to a smaller one. Be sure that your data files will fit on the smaller drive before you issue the RESIZE command!

And remember that you must have a working copy of XCOPY.COM (a very versatile copy program that is a must for any utility library) for RESIZE to work. It will look for XCOPY before running to be sure it is available.

DIETing can double your Port's disk storage!

On your Re:*Port* Disk #8, you'll find a program called DIET.EXE that can help you double your Portfolio RAM card storage space.

Of course, there are some tricks involved in getting this program to work with the Portfolio, but the results can be worthwhile.

First of all, unless you have a Port with expanded memory, you will not be able to use DIET to compress files after they are uncompressed. In order to compress the files, you will have to compress them on your desktop PC and transfer them to your RAM card via the PC Card Drive or with a parallel or serial transfer program.

Once you have the DIETed files on your RAM card, you will need to install DIET on your Portfolio in order for it to automatically uncompress the files as you use them.

This is the tricky part. You must edit your CONFIG.SYS file to include the following line:

```
DEVICE=DIET.EXE -P-K-W-M
```

Then reboot the Portfolio. You will see a message scroll quickly across the screen telling you that DIET is active.

You should now be able to read all of your compressed files.

Note that if you edit and save them, though, they will expand.

This is useful, though, especially if you have a lot of text, address books, or worksheet files that you often refer to on the road but don't ever change.

With this method, you can squeeze twice as much information onto a RAM card, thus allowing you to carry what would have been 256K on a 128K card, or 128K on a 64K card.

Note that DIET was not written by Re:*Port*, but rather is a free program.

It includes full documentation, but Re:*Port* has made a commitment to support the program. If you have any trouble figuring it out, feel free to call or write.

Beginner's Column

Help with Portfolio DOS is only a command away

In our continuing effort to persuade people to drop the File Manager program (because of its memory overhead), we'll take a look at the Portfolio's built-in DOS commands.

Did you know that you could get DOS help at any time from the Portfolio? Just type HELP (see Figure 1).

```
FILE DIR    DISK     ETC...    BATCH
copy cd     chkdsk   app off   @ do echo
del  md     fdisk    break     errorlevel
dir  rd     format   cls date  exist for
ren  path   label    prompt    goto if in
type        verify   run set   not pause
            vol      time ver  rem shift
```

Figure 1: This screen of DOS commands appears when you type HELP on the Port

The list is a bit crowded, so a few of the commands look as though they may be part of other commands, but it is still a useful screen to remind you of DOS commands that you may have forgotten (or never knew). The are conveniently arranged under five headings: File, Dir(ectory), Disk, Etc., and Batch.

Here are more detailed explanations of each command. We'll save the ones under the ''Batch'' heading for another time.

FILE COMMANDS

COPY: Copies a file to another file of a different name. Useful for copying from the internal C: drive to a RAM card in the A: drive. Example: COPY C:\MYFILE.TXT A:\COPYFILE.TXT

DEL: Deletes a file or group of files. Example: DEL A:\COPYFILE.TXT. To delete all files in a directory, type DEL *.* (the Port will make absolutely sure this is what you intend).

DIR: Types a directory listing of all the files on the current or specified drive. Example: DIR C:

REN: Renames a file. Example: REN MYFILE.TXT YOURFILE.TXT

TYPE: Displays a text file on the screen. To pause after every page, add /P to the command. Example: TYPE MYFILE.TXT /P

DIRECTORY COMMANDS

CD: Changes directory. Example: CD \SYSTEM

MD: Makes a directory. Example: MD \ADDRESS

RD: Removes a directory. The directory must be empty. Example: RD \ADDRESS

PATH: Sets up a list of directories that the Port will automatically search when executing a file. Example: PATH C:;C:\SYSTEM;A:\ADDRESS

DISK COMMANDS

CHKDSK: Displays information about your disk and memory. Example: CHKDSK C:

FDISK: Changes the size of the C: drive. Size must be a multiple of 8. Example: FDISK 8

FORMAT: Formats a RAM card. Example: FORMAT

LABEL: Gives a name to a disk. Example: LABEL REPORT

VERIFY: Determines whether DOS will verify a file when copying. Safe, but slower. Example: VERIFY ON

VOL: Displays the drive's label. Example: VOL C:

MISCELLANEOUS COMMANDS

APP: Brings up a menu of the internal applications. Or, go straight to the application you want by adding a slash and the first letter of the program. Example: APP /E (for the editor)

OFF: Turns the Port off. Example: OFF

BREAK: Determines the status of break-key checking (Fn-B). Example: BREAK ON

CLS: Clears the screen. Example: CLS

DATE: Lets you change the system date. Example: DATE

PROMPT: Changes the DOS prompt. A typical prompt includes the drive letter, the current directory, and a greater-than sign. To create this prompt, use this example: PROMPT PG

RUN: Runs a .RUN program. Example: RUN PB.RUN

SET: Sets a DOS environment variable. Not usually necessary, unless an outside program calls for it in its documentation. Example: SET VOLUME=15

TIME: Set the system time. Example: TIME

VER: The version of your Port. Example: VER

Advanced Programming

TSR programming requires some assembly

We'll cover two topics in this column: Assembly language on the Portfolio, and TSR (terminate and stay resident) programming for the Port.

In the SOURCE directory of Re:*Port* Disk #8, you will find a listing called ALARM.ASM, a heavily commented assembler source file.

The program was written on the Portfolio itself and was compiled with the A86 compiler. For more information about what ALARM.ASM does when it is compiled, see page 7.

The A86 compiler is a shareware program from Eric Isaacson, a former Intel programmer. It is available on many bulletin board systems, including CompuServe and GEnie. You can order a registered copy from Isaacson for $50 by writing to:

Eric Isaacson
416 E. University Ave.
Bloomington, IN 47401-4739

A86 is a 23K program that combines the functionality of a compiler and a linker. If you are familiar with MASM from Microsoft, then you should have no problem with A86.

The best thing about A86 is that it will work on the Portfolio itself, so you can directly compile and link your programs on the Port. You write the program using the Editor, then compile it with A86. If there are errors, your original source code file is updated with error messages.

If you're going to be writing assembly language programs for the Portfolio, go with A86!

Now that we know the best way to get assembly language onto the Port, the next question is "Why?"

ON DISK

Source code to Re:*Port* programs can be found on Re:*Port* disk #8 in the \SOURCE directory

There are several answers. I am not well-versed in assembly language, but if you are, that could be an answer in itself.

Other reasons are that assembly language tends to produce very tight programs, which is obviously good given the Port's limited storage space. Assembly language is also often the best way to access hardware interrupts and to write directly to hardware ports, which you'll do a lot as you get more advanced in Portfolio programming. Finally, assembly language is probably the best way to go if you're writing TSR programs for the Portfolio.

TSR programs generally as used to change some global function of the computer. For instance, ALARM.ASM changes the sound of the Portfolio's alarm into a user-defined song. Because this is the kind of change that we will want to have running all the time, it needs to be a TSR program so it can stop execution yet stay in memory waiting to be called on again, without the user having to get involved beyond typing in the program's name one time.

This is accomplished by writing what is called an interrupt handler routine.

The Portfolio (and all DOS machines) uses interrupts to control machine functions "in the background" as the user works on the current program. The Portfolio stores a list of these interrupts' addresses in memory, and whenever one of the interrupts is needed (for instance, to sound an alarm), it looks at this list, finds the address of the interrupt, transfers control to that program, and then returns to what the user was doing.

The trick is to change the address of the built-in interrupt handler to the address of your interrupt handler. That is actually quite simple. The addresses for all interrupts are stored at memory location 0000:0000 plus the interrupt number times four.

Be sure to save the original values before you change them. You will need to transfer control to the original interrupt before or after your interrupt into order to fool the computer into thinking that nothing happened! See ALARM.ASM for examples.

In general, you will want to write your TSR portion of the program at the beginning of your source code. Write the non-TSR portion last. This ensures that the part that stays in memory will fall to the lowest possible area, leaving more RAM available for other programs.

Most interrupts are told what to do by passing values in the AX register. AH typically holds the function number, and AL holds the subfunction. In ALARM.ASM, you can see that it checks AH for $15, which is the function for creating sound, and then checks AL for $02, which is the alarm sound subfunction.

We'll continue with a more in-depth look at TSR programs as we deliver more examples next time!

Online Services

Exciting new uploads available on CIS

The past two months have seen the upload of two great new Portfolio programs on the CompuServe APORTFOLIO forum. They are PORTMON and SERMON, two debugger programs adapted for the Portfolio.

Programmer BJ Gleason converted these two programs so that they would fit on the Port's smaller screen. If you're familiar with the DEBUG command in DOS, then you will love these programs.

PORTMON works much like DEBUG, but lacks some features such as an a(ssemble) command. SERMON allows you to debug Portfolio programs via the serial interface, if you are working on a program with graphics displays.

Both programs are definitely worth the download!

On GEnie, there are several transcripts of the weekly roundtable discussion available to download that contain answers to frequently asked Portfolio questions.

ON DISK

Catalogues of the new files available on CompuServe and GEnie are available on your Re:*Port* Disk #8 in the \ONLINE directory.

CompuServe wants all Re:*Port* readers who aren't members to sign up now, so they've put forward a special offer to our subscribers.

You can get a sign-up kit and $15 worth of free connect time by calling (800) 848-8199 (ask for Operator 198). Tell them that you heard about this from Re:*Port*, an APORTFOLIO forum member.

Also, a list of Portfolio files that are available on the free Internet computer network is online at GEnie. If you have Internet access, you may want to check it out.

Contest

Win a 64K Card to increase your storage space

The winner of contest #7, and a $5 gift certificate, is Peter Mulligan of King of Prussia, PA. Congratulations!

As you may remember, we asked for more participation in the last contest, and we received it. But we still aren't getting anywhere near half of the subscribers to enter.

OK, maybe the prizes haven't been a big draw. A $5 gift certificate is good, but it isn't great. How about an actual piece of Portfolio merchandise -- like a 64K RAM card, an $80.96 value for Re:*Port* subscribers!

As usual, you may enter through the mail, over the phone, or via CompuServe or GEnie, but you must enter by Dec. 15, 1992. And you also must provide all of the information asked for on the form above. Please enter, as this helps us determine what kind of programs we need to focus on for future issues.

Re:*Port* Contest #8

Name: ____________________

Favorite use of the Portfolio: ________

____

Favorite application (pick two):

Editor   Worksheet   Address Book

Diary   Calculator

To enter, write:
Re:*Port* Contest
806-34 Benton Drive
Iowa City, IA 52246-5207

Or call, 1-319-354-6893 and leave a message.

Or send Email to 73770,2021 on CompuServe; leave it to DAVESTEWART on GEnie.

Mailbox

Dear Re:*Port*,

I did not know if you knew this yet. Your Port checkbook program (CHECK.EXE, included with Vol. 1, Issue 3) does work with Quicken 5.0 and Quicken for Windows.

Quicken defaults on its Import feature to a *.qif file. I just change this to *.txt, insert my ram card in the ram card drive and away we go. I import d:check.txt.

The only thing that I have to do is add my categories and classes. I don't remember these and they are loaded into Quicken's memorized transactions.

In any event, the program helps me keep track of my checkbook and not have to type the info twice.

Thought you would like to know.

--Craig Davis
On CompuServe

[Editor's reply] Thank you! I did not know that CHECK.EXE would work with any version of Quicken above Version 3. With your tip, current Quicken users can take advantage of the Port's ability to keep track of financial records while you are away from the desk.

Of course, Re:*Port* appreciates suggestions and tips on how to use any Portfolio programs better, including its own!

Dear Re:*Port*,

Thanks for sending my first Re:*Port* newsletter. It's exactly the thing I was looking for -- an information paper exclusively for my favorite ''toy.''

The most interesting thing about the last issue is, of course, the article about Optrols' Flash Memory Cards. I am very interested in this new technology and will surely order one or two of the cards as soon as they are available.

I already own a ''Memory Module'' from the German distributer Becker & Partner. It is a kind of extended RAM card with a bigger battery. Unfortunately, this model does not work to my contentment, so I was glad to read about a new kind of ''hard disk'' for the Portfolio.

As I already told you, I am very interested in earlier issues of Re:*Port* and some of your programs, especially Macro-Maker.

Please send me some information on which issues of the newsletter are still available.

--Wolf-Theo Holl
Germany

[Editor's reply] The Optrol Flash cards did create quite a bit of excitement in the Portfolio community, and it is a shame that they will not exist anytime soon (see Page 3).

The Becker & Partner cards, mentioned in an earlier issue of Re:*Port*, are a more expensive alternative. I would be interested to hear more about what makes you unhappy with them, so our readers can make a more balanced judgment before buying one based solely on the amount of storage available.

As for back issues, see Page 2 for a complete list.

Dear Re:*Port*,

Thanks for the gift certificate for resubscribing! You offer a great service.

For many years before owning a Portfolio, I used dBase to keep track of my addresses and phone numbers. To keep from having to start up the computer just to call a friend, I printed most of the information onto continuouus rolodex cards, updating the cards as information changed. This also allowed me to keep two current rolodexes at work and home: Each time a record was changed, I printed two copies of it.

I haven't yet been able to figure out how to print out the Portfolio's address records on single cards and to selectivly print only the cards that have changed. (In dBase, I had a print field that I would switch on as I edited the record.)

Is there a way to selectively print the Portfolio's address records onto cards? If not, is there a program that will do so?

--Steve Sprague
Washington

[Editor's reply] There is a way, but it is not as direct as what you were doing in dBase.

Every time you update a record in the Address Book, you move the cursor to the top of that record, hit Fn-3, M (for Mark), then move the cursor to the bottom of the record and hit Fn-3, S (for Store). Then, hit the Atari-E key combination to call up the built-in Editor program.

Once in the editor, press Fn-3, P (for Paste). This will paste the entire new address record into an Editor file. Repeat the process as needed, then print the editor file onto rolodex cards.

Be sure that you have the Clipboard Save function turned on to do this. That's in the Setup menu, under Applications.